Coevolutionary Computation for Synthesis of Recognition Systems

Krzysztof Krawiec[1] and Bir Bhanu
Center for Research in Intelligent Systems
University of California, Riverside, CA 92521-0425, USA
{kkrawiec,bhanu}@cris.ucr.edu

Abstract

This paper introduces a novel visual learning method that involves cooperative coevolution and linear genetic programming. Given exclusively training images, the evolutionary learning algorithm induces a set of sophisticated feature extraction agents represented in a procedural way. The proposed method incorporates only general vision-related background knowledge and does not require any task-specific information. The paper describes the learning algorithm, provides a firm rationale for its design, and proves its competitiveness with the human-designed recognition systems in an extensive experimental evaluation, on the demanding real-world task of object recognition in synthetic aperture radar (SAR) imagery.

1. Introduction

Currently, most real-world recognition systems are handcrafted and make intense use of task-specific domain knowledge to attain a competitive performance level. This makes their design resource-demanding and prohibits the transferability of knowledge between different applications.

In this paper, we propose a general-purpose visual learning method that requires only general background knowledge related to Computer Vision (CV) and Pattern Recognition (PR). The method automates the design process and does not make assumptions about the training data. To cope with the complexity of synthesis task, we break it down into components and employ cooperative coevolution [15] to perform iterative co-design, as it does not require the explicit specification of objectives for each component.

2. Motivations

2.1. Rationale for learning in CV

Despite the enormous body of literature produced in the Computer Vision field during the past two decades, the progress in overall understanding of design principles, reduction of design costs, and increase of performance for real-world CV systems for 3D object recognition has been limited. The proposed methods are often over-specialized, make demanding assumptions, and are non-transferable between different application areas. The tedious, costly, and highly subjective trial-and-error approach, is in fact still the dominant method for designing real-world object recognition systems.

We attribute this state of affairs to the fact that not enough attention has been paid to high-level learning methods for synthesis and design support of object recognition systems. Few interesting contributions (see Section 3) that pursued this direction, are rather exceptions. This negative trend may be overturned only by incorporating *adaptive learning* methodologies, which meet the demands of real world tasks, do not assume much about the input data, and focus on *optimality of the entire recognition system*. This becomes a necessity as the complexity of real-world tasks increases and/or the tasks themselves change with time.

2.2. Rationale for evolutionary learning in CV/PR

The paradigm of Evolutionary Computation (EC) has found a significant number of applications in image processing and analysis. It has been found effective for its ability to perform global parallel search in high-dimensional spaces and to resist the local optima problem. However, in most approaches the adaptation is limited to parameter optimization; here, we take a further step and synthesize entire feature extraction procedures.

The proposed approach follows the paradigm of *feature-based* recognition, so the evolutionary search proceeds in the space of image features. In particular, if we define the feature extraction module as a graph of interconnected building blocks, which encapsulate basic operations, then the task of synthesis (learning) of recognition system consists in manipulating such graphs. Given no other extra source of knowledge, the structure of this exponential-sized search space is unknown *a priori*

[1] On a temporary leave from Institute of Computing Science, Poznań University of Technology, Poznań, Poland.

and it may be effectively searched only by means of metaheuristics. Evolutionary computation is nowadays the most intensively investigated and state-of-art metaheuristics, with a long record of success stories in optimization and learning, including attaining human-competitive performance and outperforming patented solutions [9].

2.3. Rationale for coevolutionary learning in CV/PR

It is widely recognized that the scalability of basic learning methods with respect to the complexity of the task is usually limited; e.g., one can successfully train a neural network to recognize characters, but this result cannot be extrapolated to, for instance, interpretation of outdoor scenes. This observation motivated many recent research trends in related disciplines, e.g., multi-agent systems in Artificial Intelligence, ensembles of classifiers in Machine Learning, multiple classifiers in Pattern Recognition, etc. Results obtained there clearly indicate that further progress cannot be made without resorting to some higher-level learning methodologies that inherently involve *modularity* (see, e.g., [4]).

To provide modularity, we use *Cooperative Coevolution* (CC) [15], which, besides being appealing from the theoretical viewpoint, has been reported to yield interesting results in some experiments [19]. As opposed to standard EC which maintains a single population of individuals, each being a complete solution to the problem, in CC $n>1$ populations are co-evolved, with each individual encoding only a *part* of the solution. To undergo evaluation, individuals have to be (temporarily) combined with individuals from the remaining populations to form a solution. This joint evaluation scheme forces the populations to cooperate. In [2], we provided an experimental evidence for the superiority of CC-based feature construction over EC approach in the standard machine learning setting; here, we extend this idea to visual learning.

According to Wolpert's 'No Free Lunch' theorem [21], the choice of coevolution as a particular search method is irrelevant, as the average performance of any metaheuristic search over a set of all possible fitness functions is the same. In real world, however, not all fitness functions are equally probable. Most real-world problems are characterized by some features that make them specific. The practical utility of a search/learning algorithm depends, therefore, on its ability to detect and benefit from those features.

The *decomposable nature* of visual learning task is such a feature, and CC seems to fit it well, as it provides an elegant and general framework for iterative co-design of components. It allows for breaking up a complex problem into components *without specifying explicitly the objectives for them*. The manner in which the individuals from populations cooperate *emerges* as the evolution proceeds. This makes CC, in our opinion, a perfect match for CV/PR, where it is very often the case that defining an objective (function) for particular intermediate stage of processing is difficult.

3. Related work and contributions

The area of visual learning, especially high-level learning aimed at synthesis of recognition systems, received only scant attention in mainstream research of CV/PR. Nevertheless, some past contributions pointed out interesting research directions, involving blackboard architecture [5], case-based reasoning, reinforcement learning [13][14], and evolutionary computation [7][16] [18], to mention the most predominant. In particular, [16] describes an approach that is comparable to that presented in this paper (however, without engaging coevolution) and tests it on a similar data.

This paper reports the *human-competitive performance of a synthesized recognition system* on a non-trivial recognition task. The major contribution is a general method that, given exclusively a set of training images, performs visual learning and yields a complete feature-based recognition system. Its novelty consists mostly in (i) *procedural representation* of features for recognition, (ii) utilization of *coevolutionary computation* for induction of image representation, and (iii) *learning process* that synthesizes features prior to classifier induction.

4. Technical approach

4.1. Coevolutionary synthesis of features

Following the feature-based recognition paradigm, we split the object recognition process into two modules: *feature extraction* and *decision making*. The algorithm learns from a finite set of training examples (images) D in a *supervised* manner, i.e. requires D to be partitioned into mutually disjoint decision classes D_i.

The visual learning consists here in a CC-based search performed in the space of feature extraction procedures. The cooperation may be characterized as taking place at *feature level*, as it aims at building the complete image *representation* (feature vector $\mathbf{Y}$), with each population responsible for evolving some of those features. In particular, each individual I from given population encodes a single *feature extraction procedure* that, given an input image X, produces vector of features $\mathbf{y}(I,X)$. For clarity, details of encoding of a single feature extraction procedures are provided in Section 4.2.

The coevolutionary search proceeds in all populations independently, except for the evaluation phase, shown in Fig. 1. To evaluate an individual I_j from population #j, we first provide for the remaining part of the representation. For this purpose, representatives I_i^* are selected from all the remaining populations $i \neq j$. A representative I_i^* of i^{th} population is defined here in a way that has been reported to work best [19]: it is the best individual w.r.t. the previous evaluation. In the first generation of evolutionary run, since no prior evaluation data is given, it is a randomly chosen individual.

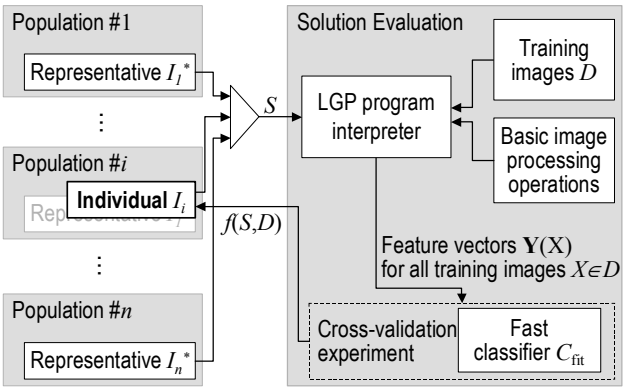

Fig. 1. **The evaluation of an individual I_i from i^{th} population.**

Subsequently, I_j is temporarily combined with representatives of all the remaining populations to form the solution

$$S = \left\langle I_1^*, \ldots, I_{j-1}^*, I_j, I_{j+1}^*, \ldots, I_n^* \right\rangle .$$

Then, the feature extraction procedures encoded by individuals from S are 'run' (see Section 4.2) for all images X from the training set D. The feature values $\mathbf{y}$ computed by them are concatenated, building the compound feature vector $\mathbf{Y}$:

$$\mathbf{Y}(X) = \left\langle \mathbf{y}(I_1^*, X), \ldots, \mathbf{y}(I_j, X), \ldots, \mathbf{y}(I_n^*, X) \right\rangle .$$

Feature vectors $\mathbf{Y}(X)$, computed for all training images $X \in D$, together with the images' decision class labels constitute the dataset:

$$\left\langle \mathbf{Y}(X), i : \forall X \in D_i, \forall D_i \right\rangle .$$

Finally, cross-validation, i.e. multiple train-and-test procedure is carried out on these data. For the sake of speed, we use here a fast classifier C_{fit} that is usually much simpler than the classifier used in the final recognition system. The resulting predictive recognition ratio becomes the evaluation of the solution S, which is subsequently assigned as the fitness value $f(\,)$ to the individual I_j, concluding its evaluation process:

$$f(I_j, D) = f(S, D) = \left| \left\{ \left\langle \mathbf{Y}(X), i \right\rangle, \forall D_i, \forall X \in D_i : C_{fit}(\mathbf{Y}(X)) = i \right\} \right| / \left| D \right| .$$

Using this evaluation procedure, the coevolutionary search proceeds until some stopping criterion (usually considering computation time) is met. The final outcome of the coevolutionary run is the best found solution/representation S^*.

4.2. Representation of feature extraction procedures

In the proposed approach, each individual encodes a feature extraction procedure that is a sequence of predefined basic operations. For this encoding, we adopt a variety of Linear Genetic Programming (LGP) [1], a hybrid of genetic algorithms (GA) and genetic programming (GP) [7]. The individual's genome is a fixed-length string of bytes, representing a sequential program composed of (possibly parameterized) basic *operations* that work on images and scalar data. This representation combines advantages of both GP and GA, being both procedural and more resistant to the destructive effect of crossover that occurs in 'regular' GP.

A feature extraction procedure accepts an image X as input and yields a vector $\mathbf{y}$ of scalar values as the result. Its operations are effectively calls to image processing and feature extraction functions, which work on *registers*, and may use them for both input as well as output arguments. *Image registers* store processed images, whereas *real-number registers* keep intermediate scalar results and features. Each image register has single channel (grayscale), the same dimensions as the input image X, and maintains a rectangular *mask* that, when activated by an operation, limits the processing to its area. For simplicity, the numbers of both types of registers are controlled by the same parameter m.

Each chunk of 4 consecutive bytes in genome encodes a single operation with the following components: (a) operation code, (b) mask flag – decides whether the operation should be global (work on the entire image) or local (limited to the mask), (c) mask dimensions (ignored if the mask flag is 'off'), and (d) arguments: references to registers to fetch input data and store the result. For parametric operations, selected arguments stored in scalar registers are interpreted as parameters. An exemplary encoded operation could be: morphological opening, applied locally to the mask of size 14×14 to the image fetched from image register pointed by argument #1, and storing the result in image register pointed by argument #2.

This way of encoding gives the learning process the possibility of manipulating the procedure at three levels of abstraction: (i) choice of operations, (ii) parameter setting, and (iii) data flow. There are currently 70 operations implemented in the system. They mostly consist of calls to functions from Intel Image Processing [6] and OpenCV

[12] libraries, and encompass general-purpose image operations (processing, basic features, geometrical moments, image statistics), mask-related operations, and elementary arithmetic and logic. The processor-optimized implementation of libraries provides short processing times of operations and good scalability w.r.t. image size.

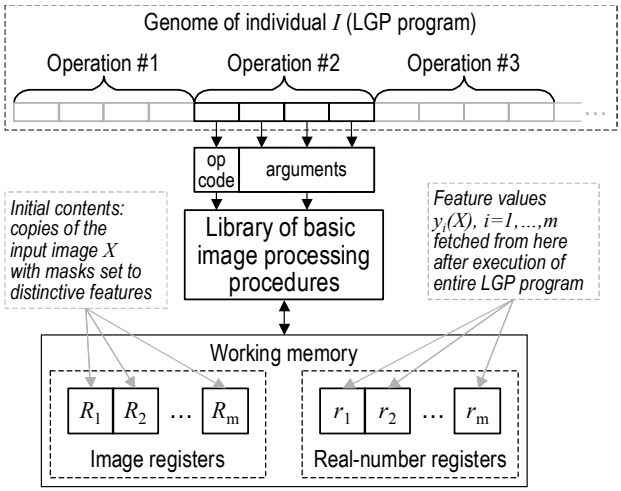

Fig. 2. Decoding and execution of LGP code contained in individual's *I* genome (for a single image *X*).

The processing of a single input image $X \in D$ by the LGP procedure encoded in an individual I starts with initialization of each of the m image registers with X (Fig. 2). The masks of images are set to m most distinctive local features (here: bright 'blobs') found in the image. Real-number registers are set to the center coordinates of corresponding masks. Then, the operations encoded by I are carried out one by one, and intermediate results are stored in registers. At the end of execution, the scalar values $y_j(I,X)$, $j=1,\ldots,m$, contained in the m real-value registers are interpreted as the output yielded by I for image X. The values are gathered to form an individual's output vector

$$\mathbf{y}(I,X) = \langle y_1(I,X),\ldots,y_m(I,X)\rangle,$$

that is subject to further processing (see Section 4.1).

4.3. Architecture of the recognition system

The overall recognition system consists of (i) best feature extraction procedures S^* constructed using the approach described in Sections 4.1 and 4.2, and (ii) classifiers trained using those features.

Facing the suboptimal character of representations elaborated by the evolutionary process, we incorporate multi-agent methodology that aims to compensate for that imperfectness and allows us to boost the overall performance. The basic prerequisite for the agents' fusion to become beneficial is their *diversification*. This may be ensured by using homogenous agents with different parameter setting, homogenous agents with different training data, heterogeneous agents, etc. Here, the diversification is naturally provided by the random nature of the genetic search. In particular, we run *many* genetic searches that start from different initial states (initial populations). The best representation S^* evolved in each run becomes a part of a single *subsystem* in the recognition system's architecture. Each subsystem has two major components: (i) a representation S^*, and (ii) a classifier **C** trained using that representation. As this classifier training is done once per subsystem, a more sophisticated classifier **C** may be used here (as compared to the classifier C_{fit} used in the evaluation function).

The subsystems process the input image X independently and output recognition decisions that are further aggregated by a simple majority voting procedure into the final decision. The subsystems are therefore homogenous as far as the structure is concerned; they only differ in the features extracted from the input image and the decisions made.

5. Experimental results

The primary objective of the experiment was to synthesize, using the described methodology, an object recognition system for a non-trivial CV/PR task. Secondary goals consisted in testing the scalability of the approach with respect to the number of decision classes and the number of subsystems n_{sub} used in the voting procedure.

As experimental testbed, we chose the demanding task of object recognition in synthetic aperture radar (SAR) images. The difficulties that make recognition in this modality extremely hard include poor visibility of objects (usually only prominent scattering centers are visible), low persistence of features under rotation (azimuth change), and high levels of noise. The source of images was MSTAR public database [17] containing real images of several objects taken at different poses (azimuths) and at 1-foot spatial resolution. From the original complex (2-channel) SAR images, we extracted the magnitude component and cropped it to 48×48 pixels. No other form of preprocessing has been applied.

The following parameter settings have been used for each coevolutionary run: selection operator: tournament selection with tournament pool size = 5; mutation operator: one-point, probability 0.1; crossover operator: one-point, probability 1.0; number of registers (image and numeric) m: 2; number of populations n: 4; genome length: 40 bytes (10 operations); single population size: 200 individuals. The remaining parameters have been set to defaults used in ECJ software package [11].

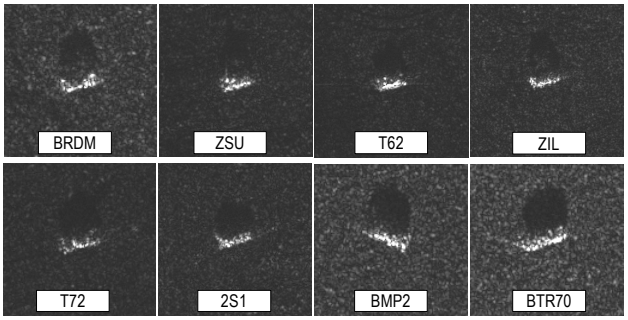

Fig. 3. Selected objects and their SAR images used in the learning experiment.

To estimate the performance of the learning process in a limited period of time, each evolutionary run has been limited to 4000 seconds on PC with Pentium 1.4 GHz processor. Decision tree inducer C4.5 (as implemented in WEKA [20]) has been used as classifier C_{fit} for feature set evaluation during evolutionary run. This choice was motivated by the low computational complexity of C4.5's induction and querying. However, the final synthesized recognition system uses support vector machine with polynomial kernels of degree 3 as classifier **C**, with complexity parameter set to 10 [20]. This classifier is much more time-consuming in learning, but provides significantly better predictive accuracy.

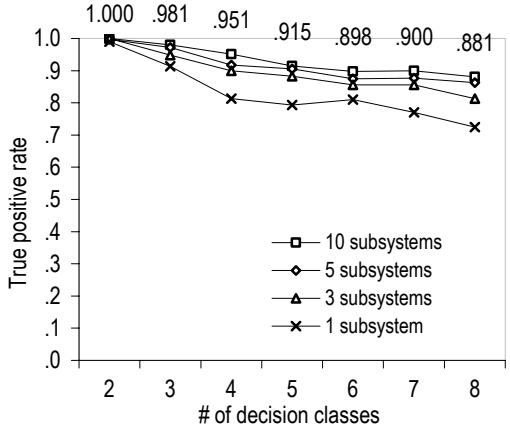

Fig. 4. **Test set recognition ratio as a function of number of decision classes for different number of subsystems n_{sub}.**

To investigate the scalability of the proposed approach with respect to the problem size, we prepared several datasets with increasing number of decision classes, using 15-deg. depression (elevation) angle data. The smallest problem contains l=2 decision classes: BRDM and ZSU. Consecutive tasks were created by adding the decision classes up to l=8 in the following order: T62, ZIL, T72, 2S1, BMP2, and BTR70 (see Fig. 3). For i^{th} decision class, its representation D_i in the training data D consists of 120 images, sampled uniformly with respect to azimuth from the original MSTAR database. For l-class task, total size of training set is therefore 120**l*. The corresponding test set T contains all the remaining images (for given decision classes and depression angle) from the original MSTAR collection.

Figure 4 presents the recognition ratio (true positive rate) as a function of the number of decision classes l represented in the training and test data, for different numbers of voting subsystems n_{sub}. It can be observed that the overall recognition performance of the system is relatively high for all the range of problems considered. When only one synthesized subsystems is used (n_{sub}=1), the recognition performance decreases significantly as new decision classes are added to the problem. However, using more voters rapidly improves the scalability, and with n_{sub}=10 voters, new decision classes do not affect the performance much. Note that in the test, we have not used 'confusers', i.e. test images from different classes that those present in the training set.

6. Conclusions

In this contribution, we provided evidence for the possibility of synthesizing, without or with little human intervention, a feature-based recognition system which recognizes 3D objects in difficult radar imagery at the performance level that is close to handcrafted, model-based solutions (cf. [3]). Let us also note that this approach attains competitive results also for recognition of object variants [10], which we, however, do not report here due to limited space.

These encouraging results have been obtained without resorting to the database of object models, without explicit estimation of object pose, without any hand-tuning of basic operations specifically to this application, and, in particular, without introducing any SAR-specific concepts or features like 'scattering center'. The learning process requires a minimum amount of training data, i.e. images and their class labels only, and does not rely on domain-specific knowledge, using only general vision-related knowledge encoded in basic operations.

The graph of processing that is carried out by feature extraction procedure emerges from the evolutionary process and may be used as an interesting source of information. In particular, the evolved procedures are often novel and sometimes very different from expert's intuition, as may be seen from example shown in Figure 5. The objectivity of the learning process make the results free from subjective flaws/biases, which the human-designed solutions are prone to: in many cases, the assumption that features designed by an expert are the optimal ones has not much scientific rationale.

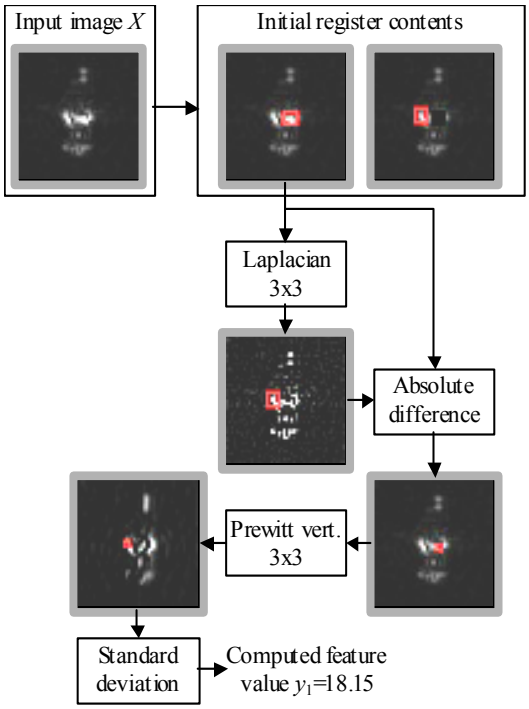

Fig. 5. An example of synthesized feature extraction procedure for BRDM image (azimuth 2.3°) as an input.

As there is no need for, usually time-consuming, matching of the recognized image with models from the database, the recognition speed of the synthesized systems is very high. The average time required by the entire recognition process for a single 48×48 image, starting from the raw image and ending up at the decision, ranged on the average from 2.2 to 20.5 ms for systems using n_{sub}=1 and n_{sub}=10 subsystems, respectively. We claim that this impressive recognition speed makes our approach suitable for real-time application.

Acknowledgements

This research was supported by the grant F33615-99-C-1440. The contents of the information do not necessarily reflect the position or policy of the U. S. Government. The first author is supported by the KBN grant no. 8T11F 006 19. We thank the authors of software packages: ECJ [11] and WEKA [20].

References

[1] W. Banzhaf, P. Nordin, R. Keller and F. Francone, *Genetic Programming: An Introduction. On the automatic Evolution of Computer Programs and its Application*. Morgan Kaufmann, 1998.

[2] B. Bhanu and K. Krawiec, "Coevolutionary construction of features for transformation of representation in machine learning," *Proc. Genetic and Evolutionary Computation Conference,* AAAI Press, New York, pp. 249-254, 2002.

[3] B. Bhanu and G. Jones, "Recognizing Target Variants and Articulations in SAR Images," *Optical Engineering*, vol. 39, no. 3, pp. 712-723, 1999.

[4] T. Dietterich, "Machine learning research: four current directions," *Artificial Intelligence*, vol. 18, no. 4, pp. 97-136, 1997.

[5] B. Draper, A. Hanson and E. Riseman. "Knowledge-Directed Vision: Control, Learning and Integration," *Proceedings of the IEEE*, vol. 11, no. 84, pp. 1625-1637, 1996.

[6] *Intel® Image Processing Library*: *Reference Manual*. Intel Corporation, 2000.

[7] M.P. Johnson, "Evolving visual routines." Master's Thesis, Massachusetts Institute of Technology, 1995.

[8] J.R. Koza, *Genetic programming - 2*. MIT Press, Cambridge, 1994.

[9] J.R. Koza, "Human-competitive Applications of Genetic Programming," *Advances in Evolutionary Computing*, A. Ghosh and S. Tsutsui, eds., pp. 663-682, Springer, 2003.

[10] K. Krawiec and B. Bhanu, "Coevolution and Linear Genetic Programming for Visual Learning," *Proc. Genetic and Evolutionary Computation Conference,* 2003 (in press).

[11] S. Luke, ECJ Evolutionary Computation System. http://www.cs.umd.edu/projects/plus/ec/ecj/, 2002.

[12] *Open Source Computer Vision Library*: *Reference Manual*. Intel Corporation, 2001.

[13] J. Peng and B. Bhanu, "Closed-Loop Object Recognition Using Reinforcement Learning," *IEEE Trans. on PAMI*, vol. 20, no. 2, pp. 139-154, Feb. 1998.

[14] J. Peng and B. Bhanu, "Delayed Reinforcement Learning for Adaptive Image Segmentation and Feature Extraction," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 28, no. 3, pp. 482-488, Aug.1998.

[15] M.A. Potter and K.A. De Jong, "Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents," *Evolutionary Computation,* no. 8, pp. 1-29, 2000.

[16] M. Rizki, M. Zmuda, and L. Tamburino, "Evolving pattern recognition systems," *IEEE Trans. Evol. Comp.*, vol. 6, no. 6, pp. 594-609, Dec. 2002.

[17] T. Ross, S. Worell, V. Velten, J. Mossing, and M. Bryant, "Standard SAR ATR Evaluation Experiments using the MSTAR Public Release Data Set," *SPIE Proceedings: Algorithms for Synthetic Aperture Radar Imagery V*, vol. 3370, pp. 566-573, April 1998.

[18] A. Teller and M.M. Veloso, "PADO: A new learning architecture for object recognition," *Symbolic Visual Learning*, K. Ikeuchi and M. Veloso, eds., pp. 77-112, Oxford Press, 1997

[19] R.P. Wiegand, W.C. Liles, and K.A. De Jong. An Empirical Analysis of Collaboration Methods in Cooperative Coevolutionary Algorithms. *Proc. Genetic and Evolutionary Computation Conference*, Morgan Kaufmann, pp. 1235-1242, 2001.

[20] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.

[21] D. Wolpert and W.G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comp.* , vol.1, no. 1, pp. 67-82, April 1997.